

## **AMENDMENTS TO THE CLAIMS:**

This listing of claims will replace all prior versions and listings of claims in the application:

1. (Currently Amended) A method for checking data consistency of data objects of distributed systems within a computer network, comprising:  
  
receiving in a second system a copy of a first data object stored in a first system;  
storing ~~a~~the copy of the first data object within the second system;  
generating a second data object from the copy of the first data object;  
sending the second data object and the copy to the first system; ~~and~~  
  
comparing at least one data package of the first data object with at least one data package of the copy of the first data object sent by the second system using a consistency check operation stored within the first data object, the second data object, or the copy of the first data object;  
  
determining, based on the comparison, whether to merge the at least one data package of the first data object and the at least one data package of the copy of the first data object; and  
  
merging the data packages based on the determination.

2. (Currently Amended) The method of claim 1, wherein the first data object and the second data object are processed according to a reaction<sub>1</sub> dependent on the consistency check operation<sub>1</sub> ~~and~~ stored within the first data object, the second data object, or the copy of the first data object.

3. (Original) The method of claim 1, further comprising storing, within the copy, a system identifier that identifies the first data object, an originating system of the first data object, or the first data object and the originating system of the first data object.

4. (Original) The method of claim 1, wherein object status information is stored within the copy.

5. (Original) The method of claim 1, wherein the data object comprises a plurality of data packages.

6. (Original) The method of claim 1, wherein the data packages are compared sequentially.

7. (Original) The method of claim 1, wherein the data packages are compared hierarchically.

8. (Currently Amended) The method of claim 1, wherein a consistency check operation description and at least one reaction are stored within the compared a data package to be compared, wherein the data package to be compared can be a data package of the first data object, the second data object, or the copy of the first data object.

9. (Original) The method of claim 1, wherein the consistency check operation compares object data included within the data packages and characterizes the data objects as consistent in the event that all object data are consistent.

10. (Original) The method of claim 1, wherein the consistency check operation compares object data included within the data packages and characterizes the data objects as consistent in the event that at least some of the object data are consistent.

11. (Original) The method of claim 1, further comprising executing a reaction in the event that that the consistency check operation characterizes the data objects as consistent, wherein the reaction includes an action selected from the group consisting of: merging the second data object into the first data object, marking the first data object for review, marking the second data object for review, marking the first data object and the second data object for review, marking the packages of the data objects for review, and replacing the first data object with the second data object.

12. (Original) The method of claim 1, further comprising executing a reaction in the event that the consistency check operation does not characterize the data objects as consistent, wherein the reaction includes an action selected from the group consisting of: canceling the second data object, and the copy of the first data object, marking the first data object for review, marking the second data object for review, marking the first data object and the second data object for review, marking the packages of the data objects for review, replacing the first data object with the second data object, and maintaining the first data object unchanged.

13. (Currently Amended) A method for checking data consistency of data objects of distributed systems within a computer network, comprising :  
sending a first data object from a first system to a second system;

storing a copy of the first data object within the first system;  
generating a second data object from the first data object;  
sending the second data object to the first system;  
comparing a data package of the copy of the first data object as originally stored with a data package of the second data object sent by the second system using a consistency check operation stored within the first data object, the second data object, or the copy of the first data object;  
determining, based on the comparison, whether to merge the data package of the second data object and the data package of the copy of the first data object; and  
merging the data packages based on the determination.

14. (Currently Amended) The method of claim 13, wherein the first data object and the second data object are processed according to a reaction, dependent on a consistency check operation, and stored within the first data object, the second data object, or the copy of the first data object.

15. (Original) The method of claim 13, further comprising storing, within the copy, a system identifier that identifies the first data object, an originating system of the first data object, or the first data object and the originating system of the first data object.

16. (Original) The method of claim 13, wherein object status information is stored within the copy.

17. (Original) The method of claim 13, wherein the data object comprises a plurality of data packages.

18. (Original) The method of claim 13, wherein the data packages are compared sequentially.

19. (Original) The method of claim 13, wherein the data packages are compared hierarchically.

20. (Currently Amended) The method of claim 13, wherein a consistency check operation description and at least one reaction are stored within ~~the compared~~ a data package to be compared, wherein the data package to be compared can be a data package of the first data object, the second data object, or the copy of the first data object..

21. (Original) The method of claim 13, wherein the consistency check operation compares object data included within the data packages and characterizes the data objects as consistent in the event that all object data are consistent.

22. (Original) The method of claim 13, wherein the consistency check operation compares object data included within the data packages and characterizes the data objects as consistent in the event that at least some of the object data are consistent.

23. (Original) The method of claim 13, further comprising executing a reaction in the event that that the consistency check operation characterizes the data objects as consistent, wherein the reaction includes an action selected from the group consisting of: merging the second data object into the first data object, marking the first data object for review, marking the second data object for review, marking the first data object and

the second data object for review, marking the packages of the data objects for review, and replacing the first data object with the second data object.

24. (Original) The method of claim 13, further comprising executing a reaction in the event that the consistency check operation does not characterize the data objects as consistent, wherein the reaction includes an action selected from the group consisting of: canceling the second data object, and the copy of the first data object, marking the first data object for review, marking the second data object for review, marking the first data object and the second data object for review, marking the packages of the data objects for review, replacing the first data object with the second data object, and maintaining the first data object unchanged.

25. (Currently Amended) A computer program product with a computer program stored thereon for checking data consistency of data objects of distributed systems within a computer network, the program comprising instructions operable to cause a processor to:

receive in a second system a copy of a first data object stored in a first system;  
store a the copy of the first data object within the second system;  
generate a second data object from the copy of the first data object;  
send the second data object and the copy of the first data object to the first system; and

compare at least one data package of the first data object with at least one data package of the copy of the first data object sent by the second system using a consistency check operation stored within the first data object, the second data object, or the copy of the first data object;

determine, based on the comparison, whether to merge the at least one data package of the first data object and the at least one data package of the copy of the first data object; and

merge the data packages based on the determination.

26. (Currently Amended) The computer program product of claim 25, the program comprising instructions operable to cause the processor to process the first data object and the second data object according to a reaction, dependent on a consistency check operation, and stored within the first data object, the second data object, or the copy of the first data object.

27. (Original) The computer program product of claim 25, the program comprising instructions operable to cause the processor to store, within the copy, a system identifier that identifies the first data object, an originating system of the first data object, or the first data object and the originating system of the first data object.

28. (Original) The computer program product of claim 25, the program comprising instructions operable to cause the processor to store object status information within the copy.

29. (Original) The computer program product of claim 25, the program comprising instructions operable to cause the processor to store data of the data object in a plurality of data packages.

30. (Original) The computer program product of claim 25, the program comprising instructions operable to cause the processor to compare the data packages sequentially.

31. (Original) The computer program product of claim 25, the program comprising instructions operable to cause the processor to compare the data packages hierarchically.

32. (Currently Amended) The computer program product of claim 25, the program comprising instructions operable to cause the processor to store a consistency check operation description and at least one reaction within ~~the compared~~ a data package to be compared, wherein the data package to be compared can be a data package of the first data object, the second data object, or the copy of the first data object.

33. (Original) The computer program product of claim 25, the program comprising instructions operable to cause the processor to utilize the consistency check operation to compare object data included within the data packages and to characterize the data objects as consistent in the event that all object data are consistent.

34. (Original) The computer program product of claim 25, the program comprising instructions operable to cause the processor to utilize the consistency check operation to compare object data included within the data packages and to characterize the data objects as consistent in the event that at least some of the object data are consistent.



35. (Original) The computer program product of claim 25, the program comprising instructions operable to cause the processor to execute a reaction in the event that the consistency check operation characterizes the data objects as consistent, wherein the reaction includes an action selected from the group consisting of: merging the second data object into the first data object, marking the first data object for review, marking the second data object for review, marking the first data object and the second data object for review, marking the packages of the data objects for review, and replacing the first data object with the second data object.

36. (Original) The computer program product of claim 25, the program comprising instructions operable to cause the processor to execute a reaction in the event that the consistency check operation does not characterize the data objects as consistent, wherein the reaction includes an action selected from the group consisting of: canceling the second data object, and the copy of the first data object, marking the first data object for review, marking the second data object for review, marking the first data object and the second data object for review, marking the packages of the data objects for review, replacing the first data object with the second data object, and maintaining the first data object unchanged.

37. (Currently Amended) A computer system for checking data consistency of data objects of distributed systems within a computer network, the system comprising:  
a first system having a first data object; ~~connected to a second system;~~  
a second system connected to the first system,  
wherein the second system receives a copy of the first data object from the first  
system, stores the copy of the first data object, generates a second data object from the

first data object, and sends the second data object and the copy of the first data object  
to the first system, which the first system being is configured to send a first data object  
to the second system, to receive a second data object and a copy of the first data object  
from the second system, and to compare compares at least one data package of the  
first data object with at least one data package of the copy of the first data object  
received by the first system using a consistency check operation stored within the first  
data object, the second data object, or the copy of the first data object[[: and]],  
determines, based on the comparison, whether to merge the at least one data package  
of the first data object and the at least one data package of the copy of the first data  
object, and merges the data packages based on the determination

~~the second system being configured to receive the first data object from the first~~  
~~system,~~

~~to store the copy of the first data object,~~

~~to generate the second data object from the first data object, and~~

~~to send the second data object and the copy of the first data object to the first~~  
~~system.~~